Accelerating Benders decomposition for short-term hydropower maintenance scheduling

Accelerating Benders decomposition for short-term hydropower maintenance scheduling

Jesús A. Rodríguez[a,b], Miguel F. Anjos[b,e], Pascal Côté[d,b], Guy Desaulniers[b,c]

[a] *HEC Montreal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Québec H3T 2A7, Canada*
[b] *GERAD, 3000 ch. de la Côte-Sainte-Catherine, Montréal, Québec H3T 2A7, Canada*
[c] *Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3A7, Canada*
[d] *Rio Tinto Aluminium, 1954 Davis, Saguenay (Quebec) Canada, G7S 4R5*
[e] *School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Rd, Edinburgh, United Kingdom, EH9 3FD*

Abstract

Maintenance of power generators is essential for reliable and efficient electricity production. Because generators under maintenance are typically inactive, optimal planning of maintenance activities must consider the impact of maintenance outages on the system operation. However, in hydropower systems finding a minimum cost maintenance schedule is a challenging optimization problem due to the uncertainty of the water inflows and the nonlinearity of the hydroelectricity production.

Motivated by an industrial application problem, we formulate the hydropower maintenance scheduling problem as a two-stage stochastic program, and we implement a parallelized Benders decomposition algorithm for its solution. We obtain convex subproblems by approximating the hydroelectricity production using linear inequalities and indicator variables, which account for the nonlinear effect of the number of active generators in the solution.

For speeding up the execution of our decomposition algorithm, we tailor and test seven techniques, including three new applications of special ordered

*Corresponding author
Email addresses: jesus.rodriguez@gerad.ca (Jesús A. Rodríguez), anjos@stanfordalumni.org (Miguel F. Anjos), pascal.cote@riotinto.ca (Pascal Côté), guy.desaulniers@gerad.ca (Guy Desaulniers)

sets, presolve and warm start for Benders acceleration. Given the large number of possible configurations of these acceleration techniques, we illustrate the application of statistical methods and computational experiments to identify the best performing configuration, which achieved a fourfold speedup of the decomposition algorithm. Results in an industrial setting confirm the high scalability on the number of scenarios of our parallelized Benders implementation.

Keywords: (R) Benders decomposition, Stochastic programming, Decomposition methods, Parallel computing, Hydroelectricity.

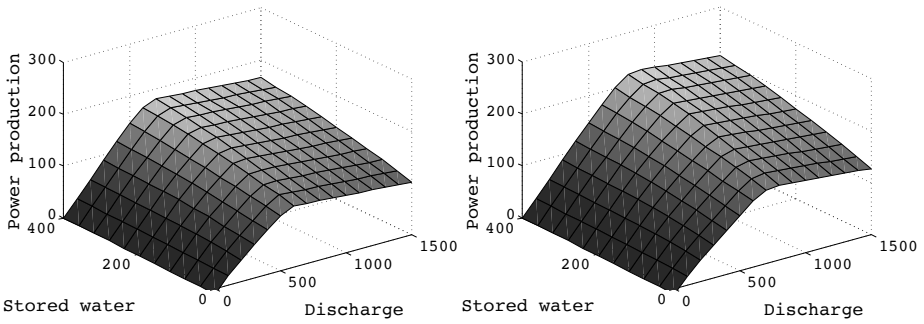
---

1. Introduction

Power producers carry out preventive maintenance activities on a regular basis to prevent costly unplanned generation outages and to extend the lifespan of the equipment. However, shutting down generators for maintenance temporarily reduces the capacity, reliability and efficiency of the system. Therefore, maintenance scheduling must anticipate the economic impact of maintenance outages. In hydropower systems this impact is difficult to estimate due to the nonlinearity of the hydroelectric generation, the uncertainty of the water inflows and the interdependence between multiple physical variables of the system.

1.1. Operational characteristics of hydropower systems

A hydropower system is composed of powerhouses with turbine-generator units driven by the potential and kinetic energy of water. In each powerhouse the hydroelectric production is a nonlinear function of the number of active generators, the turbine discharges, and the water head of the feeding reservoir or river (see Fig. 1). We refer to this function as the Hydropower Production Function (HPF). In operational hydropower problems the HPF has been represented by nonlinear functions (Finardi & da Silva, 2006; Arce, 2001; Catalão et al., 2009), non-convex piecewise linear approximations (Conejo et al., 2002; Borghetti et al., 2008), convex linear approximations (Diniz & Maceira, 2008; Rodríguez et al., 2018) and smoothing splines (Séguin et al., 2016).



Four generators Five generators

Figure 1: Hydropower Production Function in a powerhouse, for different numbers of generators and different values of turbine discharges and stored water levels.

The hydropower operation is also affected by spatial and temporal interdependencies, since water discharges can feed downstream reservoirs, and current decisions determine future costs of the system, due to the effect of the water discharges on the stored water level. Furthermore, hydroelectric generation relies on natural inflows from tributary rivers, snow-melt or rainfall which tend to be difficult to predict and can exhibit large variability (Beven, 2011).

Due to the uncertainty of the natural inflows, any current hydropower operation decision must estimate its impact on the future cost of the system, considering the possible sequences of water inflow realizations (see Fig. 2). Although stochastic dynamic programming can model this multi-stage decision problem, its application is limited by the curse of dimensionality (Bertsekas, 1995). Due to this challenge, several techniques have been applied to hydropower operation under uncertainty, such as multi-stage stochastic programming with scenario trees (Séguin et al., 2017b), progressive hedging (Carpentier et al., 2013), affine decision rules (Gauvin et al., 2017), dual dynamic programming (Pereira & Pinto, 1991; Cerisola et al., 2012) and Benders decomposition with Lagrangian relaxation (Steeger & Rebennack, 2017).

1.2. Hydropower Maintenance Scheduling (HMS)

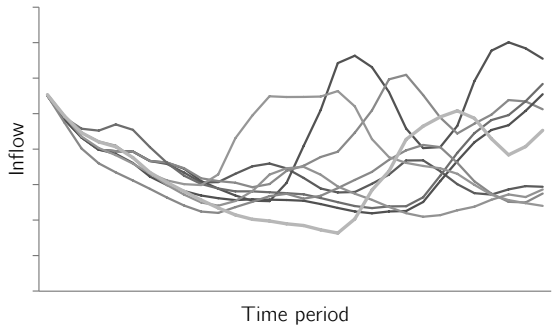


Figure 2: Scenario fan of forecasted natural inflows.

Given a list of maintenance activities to be completed within a planning horizon, the Generator Maintenance Scheduling Problem (GMS) consists in determining a calendar of maintenance outages with the best performance with respect to some criteria (e.g., profit or total cost). Such calendar must satisfy operational requirements as well as maintenance constraints, such as the time windows of maintenance activities.

In hydropower systems, GMS must consider the operational characteristics of hydroelectricity production (see Section 1.1). We refer to this problem as Hydropower Maintenance Scheduling (HMS).

Despite the extensive literature on GMS (see Froger et al. (2016) for a review), few works have focused on HMS, and some of these studies did not consider the main operational characteristics of hydropower systems (e.g. Foong et al. (2008); Canto (2008)).

For a deterministic HMS, Rodríguez et al. (2018) proposed a mixed-integer program with linear inequalities and indicator variables for approximating the three-dimensional nonlinearity of the HPF. More recently, Ge et al. (2018) and Helseth et al. (2018) applied piecewise linear approximations of the HPF, but without considering the nonlinear effect of water head and number of active generators. Given the uncertainty in HMS, Ge et al. (2018) implemented a chance-constrained formulation, whereas Helseth et al. (2018) formulated a stochastic program with maintenance decisions in the first-stage and hydropower

operation decisions in multiple stages, solved by dual dynamic programming.

Motivated by an industrial application problem, this paper presents a stochastic programming approach for HMS, incorporating a detailed approximation of the HPF. As in Rodríguez et al. (2018), we use linear inequalities and indicator variables to approximate the three-dimensional nonlinearity of the HPF with respect to the number of active generators, turbine discharges and water head. Because the resulting mathematical program is hard to solve when considering multiple inflow scenarios, we apply Benders decomposition (Benders, 1962) to partition the problem into a maintenance-only scheduling problem and scenario-wise convex operation subproblems.

Although the divide and conquer principle of Benders decomposition is a promising idea to reduce the computational effort, especially when the formulation is tight and the resulting master problem and subproblems are easy-to-solve (Magnanti & Wong, 1981), a straightforward implementation of the Benders algorithm can exhibit poor convergence and time-consuming iterations (Rahmaniani et al., 2017). Therefore, we implement seven acceleration techniques, including three new applications of presolve, warm start and special ordered sets for Benders acceleration, and we parallelize our decomposition algorithm. Using statistical methods and sequential computational experiments, we select the best combination of the implemented acceleration techniques for the decomposition method, and we compare its performance against a commercial MILP solver. For the tests, we generate instances from a real four-powerhouse system in Quebec, with up to 200 inflow scenarios.

Our Benders approach is not standard due to practical limitations in an industrial setting, regarding the access to computing facilities, commercial software and problem data. Moreover, in contrast with other applications of decomposition methods to GMS (Froger et al., 2016; Helseth et al., 2018), we focus on Benders acceleration in response to the computational challenge of accounting for the nonlinear effect of the number of active generators on the hydropower production (see Fig. 1), under uncertain inflows (Fig. 2).

The rest of this article is organized as follows: Section 2 introduces our

modeling approach. Section 3 presents the stochastic program for HMS. Section 4 introduces the Benders decomposition method and defines our problem partitioning for this method. Section 5 discusses acceleration strategies for Benders decomposition, and describes our tailored acceleration techniques. Section 6 details our methodology for selecting a best performing Benders configuration, and discusses results under various experimental conditions. Section 7 presents our summary and conclusions. Additional modeling details, results, and a nomenclature list are included in the Appendices.

2. Modeling approach

As maintenance decisions determine the set of available generators for electricity production, we compactly represent the HMS problem as

$$\max_{y \in \mathcal{Y}} Q(y) - c^{\intercal} y, \tag{1}$$

where y is a maintenance schedule vector with feasible set $\mathcal{Y}$, and c is the cost vector of the maintenance activities. The feasible set $\mathcal{Y}$ is defined by the maximum number of simultaneous outages, the time windows of maintenance activities and other relevant maintenance constraints. $Q(y)$ represents the objective value of the operational subproblem, i.e., the expected profit from hydroelectricity production during a planning horizon $\mathcal{T}$, considering operational decisions and constraints, such as turbine water discharges, which depend on the maintenance schedule y and the realization of the natural inflows.

Because the subproblem $Q(y)$ is only necessary to estimate the impact of the maintenance schedule y on the expected profit, rather than to compute the exact values of operational decisions, we relax the non-anticipacity constraints of the subproblem $Q(y)$ in order to reduce the complexity of (1). This relaxation leads to a two-stage stochastic formulation of (1), with maintenance scheduling decisions in the first stage, and senario-wise independent subproblems in the second stage (see Fig. 3). Moreover, this formulation allows us to represent the inflows uncertainty using a scenario fan of hydrological forecast sequences

(see Fig. 2), rather than a scenario tree (with a limited number of nodes for approximating the stochastic inflows).

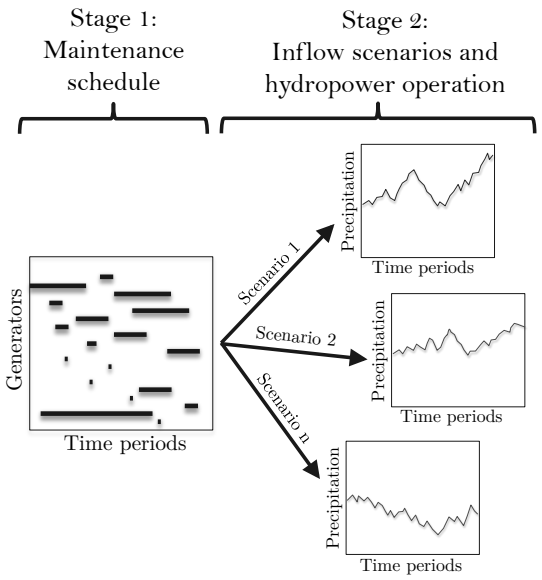


Figure 3: Two-stage stochastic approach, with maintenance decisions in the first stage and operating decisions for each inflow scenario.

In numerical experiments on short-term hydropower planning using a scenario fan formulation with decisions updated in a rolling horizon, Séguin et al. (2017a) achieved comparable solutions to those obtained from a scenario tree model, while requiring less computational effort. Although the performance of the scenario fan and the scenario tree models depend on multiple factors, such as the specific problem, its formulation and the implemented solution method, the empirical results in Séguin et al. (2017a) suggest that a scenario fan approach may be promising in some practical applications where computational times are critical.

For this work, we use a large set of inflow scenarios from Séguin et al. (2017b). These scenarios were generated by using a hydrological model with data from historical precipitations and seven-day precipitation forecasts for multiple regions in the watershed of a hydropower system in Quebec (Environment Canada, 2019).

To obtain the inflow scenarios for each region, Séguin et al. (2017b) identified a set N of historical precipitations with the closest similarity to the seven-day precipitation forecast. Then, considering that precipitations in Quebec have low correlation between consecutive days, Séguin et al. (2017b) extended the precipitation scenarios up to 30 days by appending to each precipitation series in N the historical precipitations during the same time of the year, over the past 62 years. To consider the hydrology of the watershed, the precipitation scenarios were fed into the CEQUEAU hydrological model (Charbonneau et al., 1977; INRS, 2019), which generated the corresponding inflow scenarios. See Séguin et al. (2017b,a) for more details about this methodology.

3. Two-stage stochastic program

Following the approach discussed in the previous section, we extend the MILP in Rodríguez et al. (2018) as a two-stage stochastic program for HMS. Our notation is summarized in Appendix B.

Consider a hydroelectric system with a set of powerhouses $\mathcal{I}$, and with a number of available generators $\bar{G}_{it}$ at each time period $t \in \mathcal{T}$ and powerhouse $i \in \mathcal{I}$. We assume that in each powerhouse the generators have similar characteristics. Let $\mathcal{M}$ be a list of maintenance activities to be completed within the planning horizon $\mathcal{T}$, with each activity requiring one generator outage. We define each maintenance activity m by i) the powerhouse where the activity must be executed, ii) the duration of the activity D_m, and iii) the time window $\mathcal{T}(m) \subseteq \mathcal{T}$ when the activity can initiate. Let $\mathcal{K}(i,t)$ be the set of numbers of generators that can be active at each time period and powerhouse. For determining the maintenance schedule, we define the binary variables $y_{mt} = 1$ if maintenance task $m \in \mathcal{M}$ starts at time period $t \in \mathcal{T}(m)$, 0 otherwise (2). We also define the binary variables $z_{itk} = 1$ if $k \in \mathcal{K}(i,t)$ generators are active in powerhouse $i \in \mathcal{I}$ at time period $t \in \mathcal{T}$, 0 otherwise (3).

$$y_{mt} \in \{0,1\},\ \forall\ (m,t) \in \mathcal{M} \times \mathcal{T}(m), \tag{2}$$

$$z_{itk} \in \{0,1\},\ \forall\ (i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t). \tag{3}$$

In addition, we define the following constraints that involve only first-stage maintenance decision variables:

$$\sum_{t \in \mathcal{T}(m)} y_{mt} = 1, \quad \forall\, m \in \mathcal{M}, \tag{4}$$

$$\sum_{\substack{m \in \mathcal{M}(i) \\ t' \in \mathcal{T}(m) \cap [\, t - D_m + 1,\, t\,]}} y_{mt'} = r_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{5}$$

$$r_{it} + \sum_{k \in \mathcal{K}(i,t)} k z_{itk} = \bar{G}_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{6}$$

$$\sum_{k \in \mathcal{K}(i,t)} z_{itk} = 1, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{7}$$

$$0 \leq r_{it} \leq O_{it}, \quad \forall\, (i,t) \in \mathcal{I} \times \mathcal{T}. \tag{8}$$

Constraints (4) enforce the completion of the set of maintenance activities $\mathcal{M}$ in the planning horizon $\mathcal{T}$. Constraints (5) compute the number of maintenance outages r_{it} at each time period and powerhouse. In (5) the value of r_{it} is determined by summing the variables $y_{mt'}$ corresponding to the set of activities $\mathcal{M}(i)$ in powerhouse i that could have started at time $t' \in \mathcal{T}(m)$ and still be in execution at time $t \in \mathcal{T}$ for having started in the interval $[\, t - D_m + 1,\, t\,]$.

Constraints (6) map the number of maintenance outages r_{it} into the indicator variables z_{itk} with value 1 if k generators are active at time period t and powerhouse i, and value 0 otherwise. By (7) and (3), only one z_{itk} variable is equal to one for each powerhouse and time period. Constraints (8) define the non-negativity of r_{it} and limit it to the maximum number of outages O_{it} at each time period and each powerhouse.

In addition, for the hydropower operation problem the following constraints are defined for each water inflow scenario $\omega \in \Omega$ and time period $t \in \mathcal{T}$:

$$0 \leq v_{it}, \qquad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{9}$$

$$0 \leq u_{it\omega} \leq \bar{U}_{it} \quad (\alpha^{u}_{it\omega}), \quad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{10}$$

$$\underline{S}_{it} \leq s_{it\omega} \leq \bar{S}_{it} \quad (\alpha^{s}_{it\omega}), \quad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{11}$$

$$0 \leq q^{+}_{t\omega} \leq \bar{W}^{+}_{t} \quad (\alpha^{+}_{t\omega}), \quad \forall\ (t,\omega) \in \mathcal{T} \times \Omega, \tag{12}$$

$$0 \leq q^{-}_{t\omega} \leq \bar{W}^{-}_{t} \quad (\alpha^{-}_{t\omega}), \quad \forall\ (t,\omega) \in \mathcal{T} \times \Omega, \tag{13}$$

$$\begin{aligned} s_{it\omega} - s_{i(t-1)\omega} + F\Bigg(u_{it\omega} + v_{it\omega} - \sum_{g \in \mathcal{U}(i)} (u_{gt\omega} + v_{gt\omega})\Bigg) \\ = F\xi_{it\omega} \perp \pi_{it\omega}, \quad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \end{aligned} \tag{14}$$

$$\begin{aligned} p_{itk\omega} - \beta^{u}_{h} u_{it\omega} - \beta^{s}_{h} s_{it\omega} \leq \beta^{0}_{h} \perp \gamma_{itkh\omega}, \\ \forall\,(i,t,k,h,\omega) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) \times \mathcal{H}(i,k) \times \Omega, \end{aligned} \tag{15}$$

$$\begin{aligned} 0 \leq p_{itk\omega} \leq \bar{z}_{itk} \bar{P}_{ik} \perp \lambda_{itk\omega}, \\ \forall\,(i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t), \end{aligned} \tag{16}$$

$$\sum_{k \in \mathcal{K}(i,t)} p_{itk\omega} - p_{it\omega} = 0 \perp \theta_{it\omega}, \quad \forall\ (i,t,\omega) \in \mathcal{I} \times \mathcal{T} \times \Omega, \tag{17}$$

$$\sum_{i \in I} p_{it\omega} + q^{-}_{t\omega} - q^{+}_{t\omega} = A_t \perp \psi_{t\omega}, \quad \forall\ (t,\omega) \in \mathcal{T} \times \Omega, \tag{18}$$

where $\pi_{it\omega}$, $\gamma_{itkh\omega}$, $\lambda_{itk\omega}$, $\psi_{t\omega}$ and $\theta_{it\omega}$ denote the dual variables of (14)-(17), and $\perp$ indicates their complementarity.

Constraints (9)-(13) specify the bounds of the hydropower operation decision variables: water spill $v_{it\omega}$, water discharge $u_{it\omega}$, stored water in reservoirs $s_{it\omega}$, electricity purchase $q^{-}_{t\omega}$ and electricity sale $q^{+}_{t\omega}$, respectively. In (10)-(13), we denote by α the corresponding dual variable.

Constraints (14) ensure the mass balance at each time period $t \in \mathcal{T}$ and reservoir $i \in \mathcal{I}$, considering the inflows from upstream reservoirs $g \in \mathcal{U}(i)$, as well as the uncertain natural inflows $\xi_{it\omega}$ of the respective scenario $\omega \in \Omega$. In (14), F is a scalar that converts the flow units from m^3/s to hm^3/day, for consistency of units. Moreover, in (14) we ensure a consistent solution with the initial stored water by specifying $s_{i(t-1)} = S_{i0}$ for $t = 1$.

For each powerhouse i and number of active generators k, the set of hyperplanes $\mathcal{H}(i,k)$ with parameters β_h^0, β_h^u and β_h^s in (15) define an outer approximation of the power output $p_{itk\omega}$ corresponding to values of water discharge $u_{it\omega}$ and stored water level $s_{it\omega}$, when k generators are active. For each (i,k) we compute $\mathcal{H}(i,k)$ as a subset of facets of the convex hull of the HPF, as follows: first we apply a facet enumeration algorithm to a grid of points on the surface of HPF for (i,k) (see Fig. 1); second, we sequentially remove the facet of the convex hull that yields the minimum approximation error of the power output in the remaining polyhedron, until reaching a specified number of hyperplanes. In preliminary tests with 30 hyperplanes, this approach overestimated by around 0.25% the electricity production. Such an overestimate can be reduced using an auxiliary constraint derived from a regression model (Rodríguez et al., 2018) or by increasing the number of approximation hyperplanes.

Constraints (16) restrict the generation capacity according to the number k of active generators, which is indicated by the binary variable z_{itk}. Thus, when the number of active generators is not equal to $\bar{k}$ ($z_{it\bar{k}} = 0$), the power production for this number of generators is set to zero ($p_{it\bar{k}\omega} = 0$). Constraints (17) compute the power generation $p_{it\omega}$ in each powerhouse, time period and scenario by summing the power production $p_{itk\omega}$ over the set of numbers of active generators $\mathcal{K}(i,t)$.

At each time period and scenario, the power balance is enforced by (18). In this balance, the total power injections into the system equal the power withdrawals. The injections correspond to the sum of the hydroelectric generation $p_{it\omega}$ and the electricity purchase $q^-_{t\omega}$. The power withdrawals are the electricity load A_t and the electricity sales $q^+_{t\omega}$.

Finally, the objective function of the complete problem is the sum of the expected profit from electricity trade minus the cost of maintenance activities,

$$\underset{\substack{q^+,q^-,u,v,s,\\ r,p,y,z}}{\text{maximize}} \quad \sum_{\substack{t\in\mathcal{T}\\ \omega\in\Omega}} \varphi_\omega \big(B_t^+ q^+_{t\omega} - B_t^- q^-_{t\omega}\big) - \sum_{\substack{m\in\mathcal{M}\\ t\in\mathcal{T}(m)}} C_{mt} y_{mt}, \tag{19}$$

where φ_ω is the probability of scenario $\omega \in \Omega$, C_{mt} is the cost of maintenance

activity m starting at time t, and B_t^-, B_t^+ are the electricity prices of purchase and sale at period t, respectively. Therefore, the two-stage stochastic program for the hydropower GMS is

$$\text{maximize (19) subject to (2)} - \text{(18)}. \tag{P}$$

To reduce the number of variables in (3) and the number of constraints in (15), (16) we define the set $\mathcal{K}(i,t)$ using the time windows of the maintenance activities, as proposed in Rodríguez et al. (2018) (see Appendix A.1).

4. Benders reformulation

Although several scenarios could be included into P for a richer representation of the natural inflows uncertainty (Fig. 2), the resulting large-size model would be difficult to solve in practice. Thus, we implement Benders decomposition to exploit the two-stage structure of the problem. In this section we describe our problem partitioning based on Benders decomposition, and in Section 5 we discuss and detail some acceleration strategies for this method.

4.1. The Benders decomposition method

Benders decomposition (Benders, 1962) is a solution procedure for mathematical programs with complicating variables, based on the principle that a polyhedron can be described by the convex combination of its extreme dual solutions. This method iteratively fixes candidate solutions for the complicating variables to obtain convex, easy to solve subproblems. Using the extreme dual solutions of the subproblems, the Benders algorithm generates optimality cuts (resp. feasibility cuts) that approximate the cost function (feasible space) of the subproblem into a master problem with the complicating variables (Lasdon, 1970). Since the set of extreme solutions of the subproblems is potentially large, the Benders algorithm solves a relaxed master problem (with only a subset of optimality and feasibility cuts), and sequentially includes violated cuts, as needed, until reaching a specified optimality gap. Because the Benders master

problem is a relaxation of the original (maximization) problem, any optimal value of the master problem is an upper bound of the original problem. Similarly, any feasible master problem solution $\bar{y}$ that is also feasible for the subproblems gives a lower bound LB^P for the original problem, i.e.,

$$LB^P = Q(\bar{y}) - c^\intercal \bar{y}, \tag{20}$$

where as in (1), we denote by $Q(\bar{y})$ the expected optimal value of the subproblems for solution $\bar{y}$.

4.2. Subproblem

For the application of Benders decomposition to the mathematical program P, we define a master problem with the binary variables of the problem (as defined in (2), (3)), which we compactly denote y, z. Given a master problem solution $(\bar{y}, \bar{z})$, we set $z = \bar{z}$ in (16) to obtain the scenario-wise subproblems, which for a given scenario $\omega \in \Omega$, consist in maximizing the profit of the electricity production, subject to the operational constraints, i.e.,

$$\begin{aligned} Q_\omega(\bar{z}) = \underset{q^+, q^-, u, v, s}{\text{maximize}} \quad & \sum_{t \in \mathcal{T}} (B_t^+ q_{t\omega}^+ - B_t^- q_{t\omega}^-) \\ \text{subject to} \quad & (9) - (18) \end{aligned} \tag{21}$$

To reduce the subproblem size we specify (10)-(13) as variable bounds and not as general constraints, so that they can be treated implicitly by the linear programming (LP) solver through the bounded variable simplex method. Therefore, for each bound (10)-(13) the corresponding dual variable α is not explicitly defined, and so it must be computed as the reduced cost of the corresponding bounded variable.

4.2.1. Master problem

For the mathematical program P, we define the Benders master problem with the binary variables (2), (3). This master program maximizes the expected profit of the electricity production z^{SP} minus the maintenance cost, subject to

the optimality cuts and the maintenance scheduling constraints (2)-(8). Thus, the master problem is

$$\underset{y,z,z^{SP}}{\text{maximize}} \quad z^{SP} - \sum_{\substack{m \in \mathcal{M}, \\ t \in \mathcal{T}(m)}} C_{mt} y_{mt} \tag{22}$$

subject to

$$\text{Eqs. } (2) - (8),$$

$$z^{SP} \leq \sum_{\omega \in \Omega} \varphi_{\omega} b_{\omega p}, \; \forall \; p \in \mathcal{P}, \tag{23}$$

$$z^{SP} \leq UB^{SP}, \tag{24}$$

where (23) are the optimality cuts corresponding to a set $\mathcal{P}$ of extreme solutions, and $b_{\omega p}$ is the cut term corresponding to a solution $p \in \mathcal{P}$, in scenario $\omega \in \Omega$. In this master problem no feasibility cuts are necessary due to the partial recourse property of P (see Appendix A.2).

As the Benders algorithm starts without optimality cuts, (24) prevents the unboundedness of the master problem at the first iteration. This constraint defines an initial upper bound UB^{SP} of the subproblem optimal value z^{SP}. Section 5.1.1 presents a method for computing UB^{SP}.

4.2.2. Optimality cuts

We compute the cut term $b_{\omega p}$ in (23) as

$$b_{\omega p} = b^{A}_{\omega p} + b^{B}_{\omega p}, \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}, \tag{25}$$

where $b^{A}_{\omega p}$ is the dual contribution of (14)-(18), and $b^{B}_{\omega p}$ is the dual contribution of the variable bounds (10)-(13).

For a given extreme solution $p \in \mathcal{P}$, $b^{A}_{\omega p}$ is the sum of the products between the right-hand side terms of (14)-(18), and their corresponding dual variables

$\pi^p_{it\omega}$, $\gamma^p_{itkh\omega}$, $\lambda^p_{itk\omega}$, $\psi^p_{t\omega}$, i.e.,

$$b^A_{\omega p} = \sum_{t \in \mathcal{T}} \Bigg(A_t \psi^p_{t\omega} + \sum_{i \in \mathcal{I}} \Bigg(F \xi_{it\omega} \pi^p_{it\omega} + \sum_{k \in \mathcal{K}(i,t)} \Bigg(z_{itk} \bar{P}_{ik} \lambda^p_{itk\omega} + \sum_{h \in \mathcal{H}(i,k)} \beta^0_h \gamma^p_{itkh\omega} \Bigg) \Bigg) \Bigg), \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}. \tag{26}$$

Notice that in (26), the terms corresponding to constraints (17) are discarded because their right-hand side is 0.

For computing $b^B_{\omega p}$, we multiply each bound by the value of the corresponding dual variable $\alpha^{pu}_{it\omega}$, $\alpha^{ps}_{it\omega}$, $\alpha^{p+}_{t\omega}$, $\alpha^{p-}_{t\omega}$ in the solution $p \in \mathcal{P}$. That is,

$$b^B_{\omega p} = \sum_{t \in \mathcal{T}} \Bigg(\bar{W}^-_t \alpha^{p-}_{t\omega} + \bar{W}^+_t \alpha^{p+}_{t\omega} + \sum_{i \in \mathcal{I}} \Bigg(\bar{U}_{it} \alpha^{pu}_{it\omega} + \bar{S}_{it} \alpha^{ps}_{it\omega} [\alpha^{ps}_{it\omega} > 0] + \underline{S}_{it} \alpha^{ps}_{it\omega} [\alpha^{ps}_{it\omega} < 0] \Bigg) \Bigg), \; \forall \, (\omega, p) \in \Omega \times \mathcal{P}. \tag{27}$$

Since the water discharge $s_{it\omega}$ has a lower bound $\underline{S}_{it}$, for the computation of $b^B_{\omega p}$ we sum either $\bar{S}_{it} \alpha^{ps}_{it\omega}$ or $\underline{S}_{it} \alpha^{ps}_{it\omega}$, depending on the sign of the corresponding dual value $\alpha^{ps}_{it\omega}$, as indicated by the Iverson brackets in (27). A positive dual value means that the upper bound is active, whereas a negative one indicates that the lower bound is binding.

5. Accelerating Benders decomposition

Because a basic implementation of Benders decomposition can be disappointing in practice (Rahmaniani et al., 2017), several works have proposed enhancements to speed up its execution. Typical acceleration techniques for Benders consist in applying special algorithms for solving the master problem or the subproblems (Magnanti & Wong, 1981; Hooker & Ottosson, 2003; Cordeau et al., 2001), relaxing the integrality conditions in the initial Benders iterations (Cordeau et al., 2001), selecting dual solutions that yield the strongest cuts (Magnanti & Wong, 1981; Papadakos, 2008), improving the approximation of the original problem in the master problem (Santoso et al., 2005; Crainic et al., 2016; Gendron et al., 2016), using Benders cuts in a branch-and-cut framework (Fortz

& Poss, 2009; Gendron et al., 2016; Fischetti et al., 2016; Cordeau et al., 2018), stabilizing the master problem solutions (Santoso et al., 2005; Ruszczyński & Świetanowski, 1997; Fischetti et al., 2016), and generating combinatorial cuts and knapsack cuts (Santoso et al., 2005; Fischetti et al., 2016; Gendron et al., 2016), among other techniques (see Rahmaniani et al. (2017) for a review). Furthermore, as the subproblems can be solved independently once the master problem solution is fixed, parallelization of the subproblems is a natural alternative for speeding up the Benders algorithm when many scenarios are considered (see e.g. Nielsen & Zenios, 1997; Linderoth & Wright, 2003). However, the efficiency of a parallel program would depend on technical aspects, such as task concurrency, data distribution and load balancing.

5.1. Implemented techniques

For speeding up our Benders implementation, we tailored and tested the following strategies: 1) MILP warm start, 2) presolve, 3) special ordered sets, 4) multi-phase relaxation, 5) valid inequalities, 6) combinatorial cuts, 7) integer rounding cuts, and 8) parallelization.

In the remaining of this section we discuss the related literature and the implementation details about these techniques.

5.1.1. MILP warm start (WS)

At each Benders iteration, we warm start the solution process of the master problem using solution information from the previous iteration. Notice that this approach differs from other *warm start strategies* in the literature, such as cut initialization (Rahmaniani et al., 2017), subproblem basis initialization (Morton, 1996; Wolf & Koberstein, 2013) or subproblem tree initialization (Hassanzadeh & Ralphs, 2014).

In a branch and bound process, the objective value of the incumbent solution (i.e., the current best feasible solution) defines a lower bound that helps to cut off sections of the branching tree with no potential of harboring an optimal solution. The tighter the cutoff value, the fewer the number of nodes to be explored in the

tree. In MILP solvers, cutoff values can be user-defined or can be computed from user-supplied initial solutions. Even if the initial solution is infeasible, MILP solvers can apply re-optimization or heuristics to obtain a new feasible solution and a corresponding cutoff value (FICO, 2107).

In the Benders algorithm, the optimal value of the incumbent solution for the original problem also defines a lower bound for the master problem. Therefore, at each Benders iteration we set a cutoff value $LB^P - \epsilon$ for the master problem, where ϵ is the default absolute optimality tolerance of the MILP solver, and LB^P is the bound computed by (20), corresponding to the incumbent solution. To further exploit the solver capabilities, at each iteration we supply the MILP solver with the master problem solution from the previous iteration.

As tightening bounds of variables can also make the search more efficient, at each iteration we define the current solution value of z^{SP} in the master problem as the upper bound UB^{SP} for the next iteration. Moreover, at the first step of the algorithm we define an initial upper bound UB^{SP} for z^{SP} in (24), computed as

$$UB^{SP} = \tilde{z}^P + c^\intercal y^0, \tag{28}$$

where $\tilde{z}^P$ is the optimal value of the linear relaxation of P, and $c^\intercal y^0$ is an upper bound on the maintenance cost.

Proposition 1. *UB^{SP} is a valid upper bound on the expected optimal value of the subproblem* (21)*,* (9)-(18)*.*

Proof. Let y^* and $\tilde{y}$ denote respectively the optimal integer solution and the linear relaxation solution of P. By the linear relaxation of P,

$$\begin{aligned} \tilde{z}^P = Q(\tilde{y}) - c^\intercal \tilde{y} &\geq Q(y^*) - c^\intercal y^* \\ &\geq Q(y^*) - c^\intercal y^0, \end{aligned} \tag{29}$$

since $c^\intercal y^0 \geq c^\intercal y^*$. Adding $c^\intercal y^0$ on each side of (29) yields $\tilde{z}^P + c^\intercal y^0 \geq Q(y^*)$, which proves that UB^{SP} is an upper bound for $Q(y^*)$. □

Notice that a value for $c^\intercal y^0$ can be obtained by maximizing the maintenance costs $c^\intercal y$, subject to the maintenance constraints (2)-(8).

5.1.2. Presolve (PS)

Using presolve routines we obtain model reductions that are valid for the master problem through all iterations. Commercial MILP solvers typically incorporate presolve routines to reduce the model size before the branch and cut procedure. Presolve operations include tightening bounds and constraints, removing redundant columns and rows, and fixing variables, based on logical implications or dual information (FICO, 2107; Bixby et al., 1999).

By reducing the domain of the variables and removing fractional solutions, presolve can improve the bounds of MILP problems (Bixby et al., 1999). However, as in Benders decomposition only part of the original problem information is included into the relaxed mater problem, the potential of presolving the master problem is reduced. Furthermore, as new rows are included at each iteration of the Benders algorithm, presolve operations such as reduced cost fixing can produce inconsistent solutions if applied to the relaxed master problem and fixed for subsequent iterations. In contrast, presolving the complete problem gives problem reductions that are valid for the master problem through all iterations. Therefore, we can accelerate the Benders algorithm with an initialization step that 1) applies to the complete problem (2)-(19) a presolve routine with all presolve operations activated, and 2) in the master problem fixes for all iterations of the Benders algorithm the binary variables that after presolving the complete problem are set to one of their bounds. Notice that the values of the variables fixed during presolve must be explicitly retrieved from the MILP solver because their values can be different from the linear relaxation solution. A similar application of presolve was only recently reported in Bonami et al. (2020), without detailing their specific presolve routines.

5.1.3. Special ordered sets (SOS)

In branch and bound algorithms, branching on sets of variables, instead of individual variables, can reduce the computational time. For this purpose, Special Ordered Sets (SOS) allow specifying sets of variables for branching decisions, based on a reference ordering value (Beale & Tomlin, 1970). Following

this idea, we apply SOS to accelerate the convergence of the Benders algorithm by using ordering information from the subproblems to guide the branching process for solving our master problem. Our approach differs from previous works that have also used SOS in Benders decomposition, but without exploiting subproblem information (Amjady & Ansari, 2014; Hazır et al., 2010),

In a SOS of type 1 (SOS-1), at most one variable may be non-zero. This set definition can represent a set of mutually exclusive ordered alternatives. When branching on a SOS-1, a variable in the ordered set is chosen, and the remaining variables in the set are fixed to zero (Beale & Tomlin, 1970).

Because the generation capacity $\bar{P}_{ik}$ increases with the number of generators k, the variables z_{itk} form a set ordered by k, $\forall\,(t,i) \in \mathcal{T} \times \mathcal{I}$. Thus, in our master problem we replace the binary condition on z_{itk} (3) with the SOS-1 definition

$$\text{SOS-1}_{it} = \{z_{itk} \rightarrow k \,:\, k \in \mathcal{K}(i,t)\}\ \forall\ (i,t) \in \{\mathcal{I} \times \mathcal{T} : |\,\mathcal{K}(i,t)\,| > 2\}, \qquad (30)$$

where the arrow symbol $\rightarrow$ indicates that k is the ordering value of the set. Since SOS work better when the cardinality of the set is not very small (FICO, 2107), we define a SOS-1_{it} only when $|\,\mathcal{K}(i,t)\,| > 2$.

Moreover, if $B_t^- \geq B_t^+ \geq 0\ \forall\ t \in \mathcal{T}$, reselling electricity cannot increase the profit. Under this condition, we include in the master problem the constraint

$$z^{SP} \leq \sum_{t \in \mathcal{T}} B_t^+ \Bigg(\sum_{\substack{i \in \mathcal{I}, \\ k \in \mathcal{K}(i,t)}} \bar{P}_{ik} z_{itk} - A_t \Bigg), \qquad (31)$$

to reinforce the order of variables z_{itk} through a bound on the expected optimal value of the subproblem, approximated by z^{SP}.

Proposition 2. *If $B_t^- \geq B_t^+ \geq 0\ \ \forall\, t \in \mathcal{T}$,* (31) *is a valid bound on the expected optimal value of the subproblem* (21)*,* (9)-(18)*.*

See Appendix A.4 for a proof of this proposition.

5.1.4. Multi-phase relaxation (MR)

Considering that the linear relaxation solution of the relaxed master problem can generate valid cuts for the original problem (Cordeau et al., 2001), we

evaluate the effect of several master problem relaxation schemes for generating initial cuts. Several authors have applied similar cut initialization methods (Rahmaniani et al., 2017).

For our master problem we define four relaxation levels of the binary variables y, z (Table 1). Among the possible sequences for applying these relaxations, we consider those that start with a complete linear relaxation (relaxation level 3) and in the subsequent phases solve an integer or partially integer master problem (relaxation levels 0, 1 or 2). To ensure a feasible solution, the last phase solves the integer master problem. We compare these relaxation sequences against a standard single-phase algorithm (without a relaxation phase, defined as sequence 0 in Table 2).

Table 1: Configuration of relaxation levels

Relaxation level	Binary variables	Linear relaxation type
0	y, z	No relaxation
1	y	Partial
2	z	Partial
3	-	Complete

Table 2: Sequences of relaxation levels for multi-phase relaxation

Index sequence	Relaxation sequence
0	0
1	3, 0
2	3, 2, 0
3	3, 1, 0
4	3, 1, 2, 0
5	3, 2, 1, 0

The linear relaxation of the master problem at the initial stages helps to

quickly generate optimality cuts. Nevertheless, to prevent an excessive number of cuts that can slow down the decomposition algorithm, we finish each relaxation stage when conditions on the number of cuts or the optimality gap of the stage are met.

5.1.5. Valid inequalities (VI)

As tight formulations can improve the convergence of the Benders decomposition algorithm (Magnanti & Wong, 1981), we test the effect of the valid inequalities (32)-(33) (Rodríguez et al., 2018) and (34) on the performance of the Benders method for the mathematical program P.

$$\sum_{\substack{m \in \mathcal{M}(i) \\ t' \in \mathcal{T}(m) \cap [\, t - D_m + 1,\, t\,]}} y_{mt'} + z_{itk} \leq 1 \tag{32}$$

$$\text{for } k = \bar{G}_{it}, \forall\ (i,m,t) \in \mathcal{I} \times \mathcal{M}(i) \times \mathcal{T},$$

$$\sum_{k \in \mathcal{K}(i,t) \setminus \{\bar{G}_{it}\}} z_{itk} \leq r_{it},\ \forall\ (i,t) \in \mathcal{I} \times \mathcal{T}, \tag{33}$$

$$r_{it} + \sum_{k \in \mathcal{K}(i,t) \setminus \{\underline{K}_{it}\}} (k - \underline{K}_{it}) z_{itk} \leq \bar{R}_{it},\ \forall\ (i,t) \in \mathcal{I}, \times \mathcal{T}, \tag{34}$$

where $\underline{K}_{it}$ and $\bar{R}_{it}$ are respectively the minimum number of active generators and the maximum number of activities simultaneously in execution at (i,t). For a derivation of (32)-(34), see Appendix A.3.

5.1.6. Combinatorial cuts (CC)

We apply Combinatorial Cuts (CC) for removing fractional solutions to the master problem. Codato & Fischetti (2006) proposed CC for removing infeasible solutions in mathematical programs with binary variables. In contrast with the Benders feasibility cuts, which are computed from the subproblem dual extreme rays, CC exclude the current binary solution $\bar{x}$ by forcing a change of value in at least one variable of $\bar{x}$. Given the variables x_j with index set $\mathcal{J}$, CC are defined as

$$\sum_{j \in \mathcal{S}} (1 - x_j) + \sum_{j \notin \mathcal{S}} x_j \geq 1, \tag{35}$$

where $\mathcal{S}$ is the set of variables in $\bar{x}$ with value 1, i.e., $\mathcal{S} = \{j \in \mathcal{J} : \bar{x}_j = 1\}$, and its complement is $\mathcal{S}' = \{j \in \mathcal{J} : \bar{x}_j = 0\}$. We obtain a stronger inequality than (35), by forcing at least one variable in each set, $\mathcal{S}$ and $\mathcal{S}'$, to have a different value, i.e,

$$\sum_{j \in \mathcal{S}} x_j \leq |\mathcal{S}| - 1, \tag{36}$$

$$\sum_{j \notin \mathcal{S}} x_j \geq 1 \tag{37}$$

Then, from (36) and (37), we obtain

$$\sum_{j \in \mathcal{S}} x_j - \sum_{j \notin \mathcal{S}} x_j \leq |\mathcal{S}| - 2. \tag{38}$$

Proposition 3. *The combinatorial cut* (38) *dominates* (35).

Proof. Inequality (35) can be rewritten as

$$\sum_{j \in \mathcal{S}} x_j - \sum_{j \notin \mathcal{S}} x_j \leq |\mathcal{S}| - 1. \tag{39}$$

As (39) and (38) have equal left-hand side, and the right-hand side of (39) is greater than the right-hand side of (38), then (38) dominates (35). □

Applying (38) to cut a suboptimal solution $\bar{y}$ to the master problem, gives

$$\sum_{(m,t) \in \mathcal{S}_y} y_{mt} - \sum_{(m,t) \notin \mathcal{S}_y} y_{mt} \leq |\mathcal{M}| - 2, \tag{40}$$

where $\mathcal{S}_y = \{(m,t) \in \mathcal{M} \times \mathcal{T}(m) : \bar{y}_{mt} = 1\}$. Notice that $|\mathcal{S}_y| = |\mathcal{M}|$, since by (4), for each activity m there is a variable $\bar{y}_{mt} = 1$.

Furthermore, when the costs of the tasks are independent of the starting time, i.e., when $C_{mt} = C_m$, $\forall\,(m,t) \in \mathcal{M} \times \mathcal{T}(m)$, different solutions $\bar{y}$ that correspond to the same solution $\bar{z}$ would have the same objective value. In this case, a valid cut is

$$\sum_{(i,t,k) \in \mathcal{S}_z} z_{itk} - \sum_{(i,t,k) \notin \mathcal{S}_z} z_{itk} \leq |\mathcal{I}||\mathcal{T}| - 2. \tag{41}$$

In (41), $\mathcal{S}_z = \{(i,t,k) \in \mathcal{I} \times \mathcal{T} \times \mathcal{K}(i,t) : \bar{z}_{itk} = 1\}$, with cardinality $\mathcal{S}_z = |\mathcal{I}||\mathcal{T}|$, since by (7), for each time period t and powerhouse i, exactly one variable $\bar{z}_{itk}$ is

equal to 1. To prevent removing optimal solutions, we only apply the cuts (40) and (41) when the objective value of $(\bar{y}, \bar{z})$ for the complete problem is lower than the cutoff value $LB^P - \epsilon$ defined in Section 5.1.1.

5.1.7. Integer rounding cuts (IRC)

Similarly, we apply integer rouding cuts for removing fractional master problem solutions. Let $a^\intercal$ and b be, respectively, the coefficient vector and the independent term of the righ-hand side of the optimality cut (23). Since the lower bound LB^P of the complete problem is also valid for the master problem, combining the bound $LB^P \leq z^{SP} - c^\intercal y$, with the optimality cut $z^{SP} \leq a^\intercal y + b$, gives the inequality $LB^P \leq (a - c)^\intercal y + b$, which can be tightened with integer rounding and division by the Greatest Common Divisor (GCD) of $\lceil a - c \rceil$ (Santoso et al., 2005; Chen et al., 2011). Thus,

$$\frac{\lceil LB^P - b \rceil}{\text{GCD}} \leq \left(\frac{\lceil a - c \rceil}{\text{GCD}} \right)^\intercal y, \tag{42}$$

is a valid cut for the master problem.

5.1.8. Parallelization

For the parallelization of the Benders algorithm, we implemented a master-slave approach, where the slave processors solve the subproblem and compute the cut terms, and the master process includes the cuts, solves the master problem and controls the execution of the algorithm (Fig. 4). The master process runs on a computer server with a MILP solver, and the slave processes run in parallel on a computational grid.

We used the Message Passing Interface (MPI) standard as a parallel programming protocol. Although the MPI standard requires explicit instructions for communications among processes, it incorporates high-performance communication routines that are suitable for our master-slave implementation.

5.2. Implementation details

Because this project was motivated by an industrial application, we adapted our implementation according to the computational resources made available by

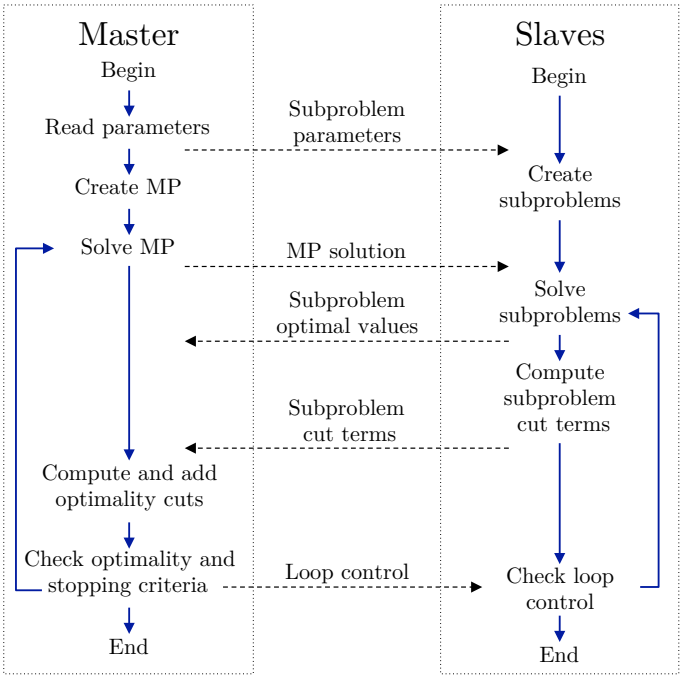


Figure 4: Simplified representation of the implemented parallel Benders algorithm

our industrial partner. The code was written in C++ with the modeling libraries Xpress BCL. The master problem was solved with the MILP solver Xpress-MP, and the subproblems were solved with the open-source solver Clp. The motivation for using an open source solver was to solve the subproblems in parallel without restrictions on the number of solver licences. For the parallelization we used the MPICH Library, which is portable, free and can use both shared and distributed memory.

In BCL, we specified the Benders optimality cuts as *delayed rows*. This cut definition is appropriate when most of the cuts are likely inactive, since only the violated cuts are reintroduced by the solver when a new solution is found. Other cuts that we implemented (valid inequalities, combinatorial cuts and integer rounding cuts) were defined in BCL as *model cuts*, which instructs the solver that these cuts can be included to remove fractional solutions. Furthermore, to avoid a large number of combinatorial cuts and integer rounding cuts, we kept only

Table 3: Basic attributes of the hydropower system. Powerhouses are ordered from upstream to downstream. Avg. inflows is the average of the forecasted natural inflows for each powerhouse, in the scenario set from Séguin et al. (2017a). See Rio Tinto (2015) for an overview of the hydropower system.

Powerhouse	Number of generators	Gen. capacity (MW)	Reserv. capacity (hm^3)	Avg. inflows (m^3/s)	
				Min.	Max.
Chute-Du-Diable	5	205	385.0	119.1	327.1
Chute-Savane	5	210	-	18.8	57.9
Isle-Maligne	12	402	4726.4	472.7	1504.5
Shipshaw	17	1587	-	0.0	0.0
Total	39	2404	5111.4	610.6	1889.5

the cuts generated in the previous iteration. The decomposition algorithm was executed in parallel on a 200-core computational grid, with one thread dedicated to each subproblem and with up to 10 threads for solving the master problem on an Intel® Xeon® 24-processor computer at 2.7 GHz, with 32.9 GB RAM.

6. Computational experiments

In this section we select the combination of acceleration techniques with the best performance on a set of test instances and we evaluate the impact of the parallelization on the computational times of the decomposition algorithm.

Our tests instances are adapted from a hydropower system of Rio Tinto in Quebec, Canada. We consider a hydro system composed of 4 powerhouses in cascade, with 39 generators, 2 reservoirs and 2404 MW of generation capacity (see Table 3). For each powerhouse and number of generators, the hydropower production function was approximated using 30 hyperplanes in (15).

6.1. Selection of acceleration techniques

Since 7 acceleration techniques of Section 5.1 can be combined in $2^7 = 128$ different ways, we are interested in identifying their best performing combination

with respect to the computational time. For this purpose, we applied the experimental methodology summarized in Fig. 5.

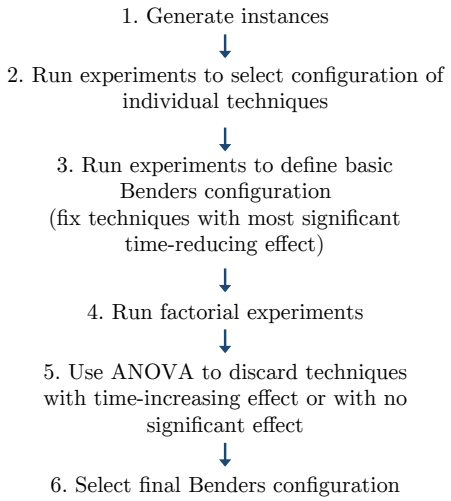


Figure 5: Experimental methodology for selecting a best performing combination of acceleration techniques

In step 1, we generated a testbed of 24 instances for the experiments. Each instance represents a HMS problem with 30 inflow scenarios, 15 time periods and 6 to 8 maintenance tasks. For each task, we randomly specified a duration between 4 to 8 days, and we defined a maintenance time window of 3 days for the starting time of each activity.

Because the computational times can differ significantly between instances, we defined as a performance metric the normalized time $\bar{t}_{jb}$ per instance

$$\bar{t}_{jb} = \frac{t_{jb} - \mu_j}{\sigma_j}, \tag{43}$$

where t_{jb} is the computational time of the instance $j \in \mathcal{J}$ on treatment $b \in \mathcal{B}$, and μ_j, σ_j are respectively, the mean and standard deviation of the computational times of instance $j \in \mathcal{J}$ in all treatments. In our experiments, a treatment corresponds to a combination of acceleration techniques or to a specific configuration of one of them.

In step 2, through preliminary experiments we identified the best configuration

for MR (relaxation sequence 4) and for VI (valid inequalities (33) and (34)). See Appendix B for details.

In step 3, we ran experiments applying each of the seven techniques individually: valid inequalities (VI), MILP warm start (WS), multi-phase relaxation (MR), special ordered sets (SOS), combinatorial cuts (CC), presolve (PS) and integer rounding cuts (IRC). As shown in Fig. 6 and Table 4, WS achieved the lowest computational times, followed by PS and SOS. Through one-sided t-tests against the basic method, we confirmed that the effect of these three acceleration techniques was highly significant on the computational time (p-value < 0.001 in Table 4). From these results, we fixed, as part of the basic configuration, the techniques with the lowest computational time (PS, SOS and WS).

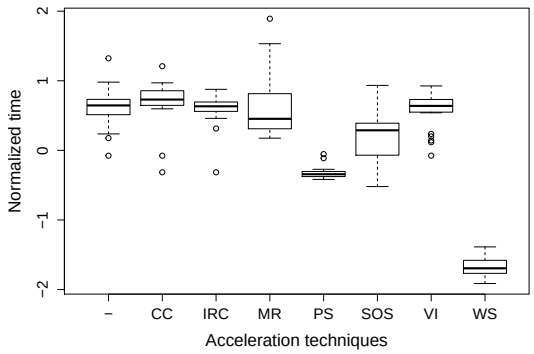


Figure 6: Boxplots of normalized computational times of 7 acceleration techniques and the basic method

For selecting the final configuration, in step 4 we ran a full factorial experiment with the remaining 4 techniques: CC, IRC, MR and VI, which corresponds to $2^4 = 16$ treatments. In step 5, an analysis of variance (ANOVA) applied to the results of this experiment indicates that CC and IRC had a significant effect (p-value < 0.05) on decreasing the computational time ($\beta < 0$), while MR had the opposite effect and VI was not statistically significant (see Table 5). Therefore, in a second ANOVA, we considered only the factors CC and IRC and their interaction term CC·IRC (Table 6). This ANOVA showed that the effects

Table 4: Summary statistics of the acceleration methods applied independently. The column *Diff.* shows the difference between the mean time of each technique and the mean time of the basic method (first row).

Treatment	Mean	Std.Dev.	Diff.	p-value
-	0.62	0.27	0.00	-
CC	0.69	0.31	0.07	0.81
IRC	0.59	0.23	−0.03	0.37
MR	0.62	0.43	0.00	0.53
PS	−0.32	0.09	−0.94	6.7e-16
SOS	0.22	0.37	−0.40	5.5e-05
VI	0.56	0.25	−0.06	0.24
WS	−1.68	0.14	−2.30	2.2e-16

of CC and IRC were statistically significant (p-value < 0.01) on reducing the computational time ($\beta < 0$). Notice that the main effects of CC and IRC (with β estimates -0.996 and -0.339, respectively) dominate interaction term CC·IRC (with β estimate 0.242), which was not statistically significant (p-value 0.169).

Table 5: Summary of linear regression model with techniques VI, MP, CC and IRC as main factors, with normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.288	0.003
VI	0.089	0.295
MP	0.428	7.6e-07
CC	−0.875	< 2e-16
IRC	−0.218	0.011

From these results, and the previously selected acceleration techniques (Table 4), in step 6 we determined that the recommended combination of the acceleration techniques for the considered problem is: PS, SOS, WS, CC and IRC. In

Table 6: Summary of linear regression model with factors CC and IRC and interaction term. Normalized computational time as response variable

	β estimate	p-value
(Intercept)	0.607	1.9e-11
CC	−0.996	1.2e-14
IRC	−0.339	0.006
CC·IRC	0.242	0.169

additional tests, this configuration achieved speedups of up to 4 times, with respect to a basic Benders implentation.

6.2. Effect of the number of scenarios

Although the representation of the inflows uncertainty can improve with the number of scenarios, in practice the model size is limited by the available computational resources and the time limit for obtaining solutions. In this section we analyze the computational time and the quality of the solutions, for different number of inflow scenarios.

6.2.1. Effect on computational time

For these experiments we consider instances with 8 maintenance tasks to be completed in a planning horizon of 15 days, for the same four-powerhouse system described at the beginning of this section. We specify a time limit of 1000 seconds for each run.

To avoid overlapping of subproblems on the available computing processors, we generate instances with up to 200 scenarios, and we assign one processor to each subproblem. The decomposition method was benchmarked against the straightforward MILP solution approach, i.e., solving model (2)-(19) with the MILP solver Xpress-MP. To observe the effect of the number of scenarios on the computational times, we kept constant all the problem parameters, except the size and composition of the set of inflow scenarios. From an initial set of

3028 scenarios from Séguin et al. (2017b), we randomly sampled 12 sets of 200 scenarios each, and we ran tests with 1, 50, 100, 150 and 200 scenarios of each set.

The results indicate that above some point between 50 to 100 scenarios, the parallel Benders decomposition with acceleration techniques outperformed the computational time of the solution with the MILP solver (Fig. 7). Furthermore, in instances with 150 and 200 scenarios, the MILP solver reached the 1000-second time limit, with average optimality gaps of 4.6 % and 6.3 %, respectively, while the Benders decomposition approach reached optimal solutions in less than 800 seconds (Fig. 7). The results also confirm that, in contrast with the straightforward MILP solution, the parallel Benders decomposition method is highly scalable on the number of scenarios. For example, between 50 to 100 scenarios the computational time of the MILP solver increased by 231.7 %, while the computational time via parallel Benders decomposition increased only by 11.5 % (Table 7).

Our results also showed that the computational times of both solution approaches (Benders decomposition and MILP solver) quickly increased with the number of maintenance tasks, and that in the tested instances the Benders method reached optimality gaps of less than 1% within 5 minutes of computation.

6.2.2. Effect on the solution quality

Results from 12 replicates for different numbers of scenarios and a given list of maintenance tasks, showed that the coefficient of variation of the optimal values tends to decrease asymptotically with the number of scenarios (see Figure 8). Naturally, this reduction of the solution variability leads to better estimates of the expected optimal value, as the number of scenarios increases.

We also conducted tests in out-of-sample scenarios to estimate gains from the stochastic solution. In these experiments we used a set of 15 instances with 4 powerhouses, 25 time periods, and 8 maintenance tasks. To execute these tests, first we randomly split the original set of 3038 inflow scenarios into two subsets: an *in-sample* scenario set for the mathematical program, and an *out-of-sample*

Table 7: Statistics on the computational times with parallel Benders decomposition and MILP-based solution, with different numbers of inflow scenarios

Number of scenarios	Benders Time		MILP Time	
	Mean	St. dev.	Mean	St. dev.
1	338.9	14.0	0.9	0.3
50	421.2	15.4	213.0	14.0
100	469.8	11.3	706.5	86.0
150	616.5	24.0	-	-
200	780.7	11.8	-	-

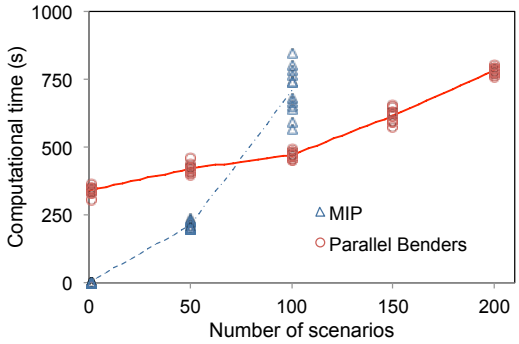


Figure 7: Computational time of solving the problem with a MILP solver and with Benders decomposition.

scenario set for computing the expected optimal values. For each instance, the first-stage solutions of the *in-sample* stochastic program (with 200 scenarios) and the integer solution of the deterministic program (with one scenario) were fixed into a set of second-stage subproblems with 200 *out-of-sample* scenarios. Finally, we computed the gain from the stochastic solution as the difference in the average optimal value between the stochastic solution and the deterministic solution in the *out-of-sample* subproblems (see Fig. 9). Whereas in eight of the tested instances the stochastic and the deterministic solutions achieved the same average *out-of-sample* optimal values, in seven instances the optimal value of the stochastic problem was more than $\$4,000$ higher than the value

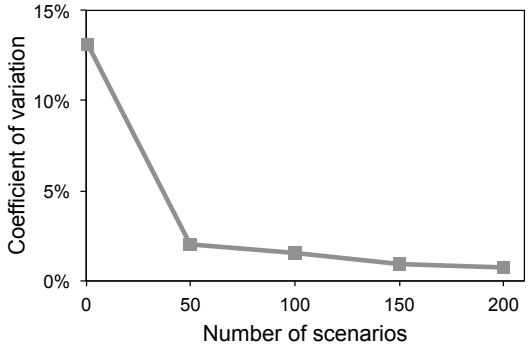


Figure 8: Variability of optimal solutions in a representative instance, with 12 replicates for each number of scenarios.

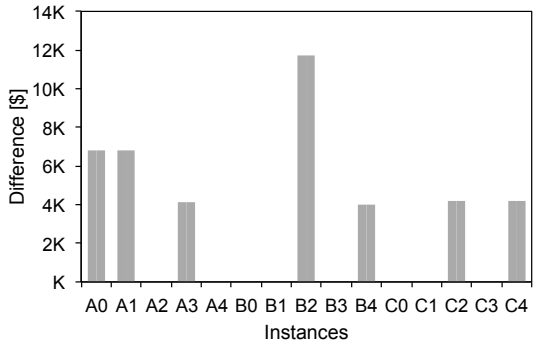


Figure 9: Absolute difference between the average *out-of-sample* optimal values of the deterministic solution and the stochastic solution, for 15 instances with 200 scenarios.

corresponding to the deterministic solution, based on an electricity price of 5 ¢/kWh, and zero direct maintenance cost, i.e., $C_{mt} = 0 \; \forall \; (m,t) \in \mathcal{M} \times \mathcal{T}(m)$. In these experiments, the average gain of $\$2,787$ over a 25-day period would be equivalent to an annualized gain of $\$40,690$. Although the expected gain from the stochastic solution is highly dependent on the specific instance (as shown in Fig. 9), this gain would be more significant under a high electricity price or a greater opportunity cost of maintenance, for instance, due to a greater number of maintenance tasks.

7. Conclusions

Given a set of forecasted inflow scenarios, we represent the hydropower maintenance scheduling problem as a two-stage stochastic program with maintenance decisions in the first stage, and hydropower operation decisions in the second stage for each water inflow scenario. This formulation approximates the three-dimensional nonlinearity of hydroelectric production by means of linear inequalities and indicator variables for each number of active generators.

To solve instances with a large number of inflow scenarios, we implemented a parallelized Benders decomposition method, and we tailored and tested seven techniques for speeding up its execution. Among these techniques, we propose new applications of presolve, special ordered sets and MILP warm start for Benders acceleration:

- In presolve we obtained model reductions that are valid for the master problem through all Benders iterations.
- By means of special ordered sets we incorporated ordering information from the subproblems to guide the branching process for solving our master problem.
- Using MILP warm start we speeded up the solution process of the master problem by reusing solution information from the previous Benders iteration. This approach differs from other *warm start strategies* in the literature, such as cut initialization or subproblem initialization (Rahmaniani et al., 2017; Morton, 1996; Wolf & Koberstein, 2013; Hassanzadeh & Ralphs, 2014).

In our experience, presolve, special ordered sets and MILP warm start were easy to implement and they yielded significant reductions in computational time.

Due to a large number of possible configurations of the seven implemented techniques, we conducted sequential computational experiments to select the combination of such techniques with the best performance. In our experiments,

the combination of presolve, special ordered sets, MILP warm start, combinatorial cuts and integer rounding cuts achieved a four-fold speedup with respect to a basic Benders implementation. In tests with up to 200 scenarios, we confirmed the high scalability of the parallelization on the number of scenarios, and the gains from the stochastic solution in *out-of-sample tests*.

Because hydropower maintenance scheduling with many maintenance tasks is still a challenging problem, future works could extend our Benders approach in a branch-and-cut framework, or they can experiment with different decomposition strategies. Future works could also incorporate further operational aspects, such as the capacity of transmission lines. Developing alternative formulations and conducting computational studies on the value of the stochastic solution are also avenues of future research.

8. Acknowledgments

We thank our anonymous reviewers for their comments and suggestions. Our thanks also to the Énergie Électrique division of Rio Tinto for their support during this project.

This research was funded by Rio Tinto, NSERC and MITACS.

References

Amjady, N., & Ansari, M. R. (2014). Non-convex security constrained optimal power flow by a new solution method composed of Benders decomposition and special ordered sets. *International Transactions on Electrical Energy Systems*, *24*, 842–857.

Arce, A. (2001). Optimal dispatch of generating units of the Itaipu hydroelectric plant. *IEEE Power Engineering Review*, *21*, 56–56.

Beale, E. M. L., & Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of

variables. In J. R. Lawrence (Ed.), *Proc. of the 5th Int. Conf. on Operations Research* (pp. 447–454). Tavistock Publications.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*, 238–252.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena scientific Belmont, MA.

Beven, K. J. (2011). *Rainfall-runoff modelling: the primer*. John Wiley & Sons.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Bixby, E. R., Fenelon, M., Gu, Z., Rothberg, E., & Wunderling, R. (1999). MIP: Theory and practice — closing the gap. In *IFIP Conference on System Modeling and Optimization* (pp. 19–49). Springer.

Bonami, P., Salvagnin, D., & Tramontani, A. (2020). Implementing automatic benders decomposition in a modern mip solver. In *International Conference on Integer Programming and Combinatorial Optimization* (pp. 78–90). Springer.

Borghetti, A., D'Ambrosio, C., Lodi, A., & Martello, S. (2008). An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *IEEE Transactions on Power Systems*, *23*, 1115–1124.

Environment Canada (2019). Meteorological Service of Canada — Open data user documentation. URL: `https://eccc-msc.github.io/open-data/readme_en/`.

Canto, S. P. (2008). Application of Benders' decomposition to power plant preventive maintenance scheduling. *European Journal of Operational Research*, *184*, 759–777.

Carpentier, P.-L., Gendreau, M., & Bastin, F. (2013). Long-term management of a hydroelectric multireservoir system under uncertainty using the progressive hedging algorithm. *Water Resources Research*, *49*, 2812–2827.

Catalão, J., Mariano, S., Mendes, V., & Ferreira, L. (2009). Scheduling of head-sensitive cascaded hydro systems: A nonlinear approach. *IEEE Transactions on Power Systems*, *24*, 337–346.

Cerisola, S., Latorre, J. M., & Ramos, A. (2012). Stochastic dual dynamic programming applied to nonconvex hydrothermal models. *European Journal of Operational Research*, *218*, 687–697.

Charbonneau, R., Fortin, J., & Morin, G. (1977). The CEQUEAU model: description and examples of its use in problems related to water resource management/Le modèle CEQUEAU: description et exemples d'utilisation dans le cadre de problèmes reliés à l'aménagement. *Hydrological Sciences Journal*, *22*, 193–202.

Chen, D.-S., Batson, R. G., & Dang, Y. (2011). *Applied integer programming: modeling and solution*. John Wiley & Sons.

Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, *54*, 756–766.

Conejo, A. J., Arroyo, J. M., Contreras, J., & Villamor, F. A. (2002). Self-scheduling of a hydro producer in a pool-based electricity market. *IEEE Transactions on Power Systems*, *17*, 1265–1272.

Cordeau, J.-F., Furini, F., & Ljubić, I. (2018). Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, .

Cordeau, J.-F., Stojković, G., Soumis, F., & Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, *35*, 375–388.

Crainic, T. G., Hewitt, M., & Rei, W. (2016). *Partial Benders decomposition strategies for two-stage stochastic integer programs*. Technical Report CIRRELT-2016-37 CIRRELT.

Diniz, A. L., & Maceira, M. E. P. (2008). A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects. *IEEE Transactions on Power Systems*, *23*, 1298–1308.

FICO (2107). Xpress-Optimizer reference manual. *Release 31.01*, .

Finardi, E. C., & da Silva, E. L. (2006). Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *IEEE Transactions on Power Systems*, *21*, 835–844.

Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Redesigning Benders decomposition for large-scale facility location. *Management Science*, *63*, 2146–2162.

Foong, W. K., Simpson, A. R., Maier, H. R., & Stolp, S. (2008). Ant colony optimization for power plant maintenance scheduling optimization-—a five-station hydropower system. *Annals of Operations Research*, *159*, 433–450.

Fortz, B., & Poss, M. (2009). An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, *37*, 359–364.

Froger, A., Gendreau, M., Mendoza, J. E., Pinson, É., & Rousseau, L.-M. (2016). Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, *251*, 695–706.

Gauvin, C., Delage, E., & Gendreau, M. (2017). Decision rule approximations for the risk averse reservoir management problem. *European Journal of Operational Research*, *261*, 317–336.

Ge, X., Xia, S., & Su, X. (2018). Mid-term integrated generation and maintenance scheduling for wind-hydro-thermal systems. *International Transactions on Electrical Energy Systems*, *28*, e2528.

Gendron, B., Scutellà, M. G., Garroppo, R. G., Nencioni, G., & Tavanti, L. (2016). A branch-and-Benders-cut method for nonlinear power design in green

wireless local area networks. *European Journal of Operational Research*, *255*, 151–162.

Hassanzadeh, A., & Ralphs, T. K. (2014). *A generalization of Benders' algorithm for two stage stochastic optimization problems with mixed integer recourse*. Technical Report Technical report, COR@L Laboratory, Lehigh University.

Hazır, Ö., Haouari, M., & Erel, E. (2010). Discrete time/cost trade-off problem: A decomposition-based solution algorithm for the budget version. *Computers Operations Research*, *37*, 649 – 655. doi:https://doi.org/10.1016/j.cor.2009.06.009.

Helseth, A., Fodstad, M., & Mo, B. (2018). Optimal hydropower maintenance scheduling in liberalized markets. *IEEE Transactions on Power Systems*, *33*, 6989–6998.

Hooker, J. N., & Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, *96*, 33–60.

INRS (2019). CEQUEAU Hydrological Model. URL: http://ete.inrs.ca/ete/publications/cequeau-hydrological-model.

Lasdon, L. S. (1970). *Optimization theory for large systems*. Courier Corporation.

Linderoth, J., & Wright, S. (2003). Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, *24*, 207–250.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, *29*, 464–484.

Morton, D. P. (1996). An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling. *Annals of Operations Research*, *64*, 211–235. URL: https://doi.org/10.1007/BF02187647. doi:10.1007/BF02187647.

Nielsen, S. S., & Zenios, S. A. (1997). Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Computing*, *23*, 1069–1088.

Papadakos, N. (2008). Practical enhancements to the Magnanti–Wong method. *Operations Research Letters*, *36*, 444–449.

Pereira, M. V., & Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, *52*, 359–375.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*, 801–817.

Rio Tinto (2015). Rio Tinto's unparalleled position on hydropower in Canada. URL: `https://youtu.be/KnwWZvn6pIU?t=147`.

Rodríguez, J. A., Anjos, M. F., Côté, P., & Desaulniers, G. (2018). MILP Formulations for generator maintenance scheduling in hydropower systems. *IEEE Transactions on Power Systems*, *33*, 6171–6180.

Ruszczyński, A., & Świetanowski, A. (1997). Accelerating the regularized decomposition method for two stage stochastic linear problems. *European Journal of Operational Research*, *101*, 328–342.

Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, *167*, 96–115.

Séguin, S., Audet, C., & Côté, P. (2017a). Scenario-tree modeling for stochastic short-term hydropower operations planning. *Journal of Water Resources Planning and Management*, *143*, 04017073. doi:`10.1061/(ASCE)WR.1943-5452.0000854`.

Séguin, S., Côté, P., & Audet, C. (2016). Self-scheduling short-term unit commitment and loading problem. *IEEE Transactions on Power Systems*, *31*, 133–142.

Séguin, S., Fleten, S.-E., Côté, P., Pichler, A., & Audet, C. (2017b). Stochastic short-term hydropower planning with inflow scenario trees. *European Journal of Operational Research*, *259*, 1156–1168.

Steeger, G., & Rebennack, S. (2017). Dynamic convexification within nested Benders decomposition using Lagrangian relaxation: An application to the strategic bidding problem. *European Journal of Operational Research*, *257*, 669–686.

Wolf, C., & Koberstein, A. (2013). Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method. *European Journal of Operational Research*, *230*, 143 – 156. doi:https://doi.org/10.1016/j.ejor.2013.04.017.

Appendix A: Model supplement

A.1 Set reduction

In Rodríguez et al. (2018), the set of numbers of generators is defined as

$$\mathcal{K}(i,t) = \left\{ k \in \mathbb{Z} : \underline{K}_{it} \leq k \leq \bar{K}_{it} \right\}, \forall (i,t) \in \mathcal{I} \times \mathcal{T} \tag{44}$$

where

$$\underline{K}_{it} = \max\{\bar{G}_{it} - O_{it}, \bar{G}_{it} - \bar{R}_{it}\}, \tag{45}$$

$$\bar{K}_{it} = \bar{G}_{it} - \underline{R}_{it}. \tag{46}$$

In (45)-(46), $\bar{G}_{it}$ denotes the maximum number of available generators at $(i,t) \in \{\mathcal{I} \times \mathcal{T}\}$, O_{it} is the maximum number of maintenance outages, and $\bar{R}_{it}$, $\underline{R}_{it}$ denote, respectively, the maximum and minimum number of activities simultaneously in execution at (i,t), according to their time windows, i.e.,

$$\underline{R}_{it} = \left| \{ (m,t) \in \mathcal{M}(i) \times \mathcal{T}(m) : L_m \leq t \leq E_m + D_m \} \right|, \tag{47}$$

$$\bar{R}_{it} = \left| \{ (m,t) \in \mathcal{M}(i) \times \mathcal{T}(m) : E_m \leq t \leq L_m + D_m \} \right|, \tag{48}$$

where for each activity $m \in \mathcal{M}$, we denote by D_m, E_m and L_m its duration, earliest starting time and latest starting time, respectively.

A.2 Conditions for feasible subproblems

From the viewpoint of computational efficiency, *complete recourse* and *partially complete recourse* are desirable properties of stochastic programming problems (Birge & Louveaux, 2011). In problems with these properties, feasibility cuts are unncecesary since the Benders decomposition method will only generate feasible solutions at each iteration. A stochastic program is said to have *complete recourse* if the second stage problem (i.e., the subproblem) is always feasible. If the stochastic program has *partially complete recourse*, the second stage problem is feasible for any feasible first stage solution and scenario realization. Following these definitions, we notice that the subproblem (21)-(13) has *partially complete recourse* (i.e., is feasible for any inflow scenario and master problem feasible solution), if the following conditions are met:

1. The system (14), (9)-(11) is feasible for any inflow realization $\xi_{it\omega}$, where $(i, t, \omega) \in \{\mathcal{I} \times \mathcal{T} \times \Omega\}$.

2. In all time periods, the electricity load A_t is not greater than the upper bound of the electricity purchase, i.e., $0 \leq A_t \leq \bar{W}_t^-, \ \forall \ t \in \mathcal{T}$.

Without loss of generality, we assume that the instances of problem P satisfy conditions 1 and 2. Notice that these conditions can be guaranteed with proper values of the variable bounds (10)-(13). If either of these conditions are not met, it would be necessary to include feasibility cuts at some iterations of the Benders algorithm. Alternatively, the partial complete recourse property can be reestablished with the introduction of artificial variables in (14), (18), and with a penalization of these variables in the objective function (21).

A.3 Valid inequalities

1. The first family of valid inequalities comes from the observation in Rodríguez et al. (2018) that in a powerhouse i, if at least one maintenance task $m \in \mathcal{M}(i)$ is in execution at time t, then the binary variable corresponding to $\bar{G}_{it}$ active generators must be equal to zero, i.e., $z_{itk} = 0$, for $k = \bar{G}_{it}$. Thus,

$$\sum_{t' \in \{\mathcal{T}(m) \,:\, (t - D_m + 1) \leq t' \leq t\}} y_{mt'} + z_{itk} \leq 1,$$
$$\text{for } k = \bar{G}_{it}, \forall \ (i, m, t) \in \mathcal{I} \times \mathcal{M}(i) \times \mathcal{T},$$

   are valid inequalities. Naturally, such inequalities are unnecessary when $\bar{K}_{it} < \bar{G}_{it}$ (44) or when the set $\{t' \in \mathcal{T}(m) : (t - D_m + 1) \leq t' \leq t\}$ is empty.

2. The second family of valid inequalities comes from the fact that for any (i, t), when the number of maintenance outages is zero, i.e., $r_{it} = 0$, then all $\bar{G}_{it}$ generators are active ($z_{itk} = 1$, for $k = \bar{G}_{it}$) (Rodríguez et al., 2018). By (7), it follows that $z_{itk} = 0$ for $k < \bar{G}_{it}$, which is equivalent to

$$\sum_{k \in \mathcal{K}(i,t) \setminus \{\bar{G}_{it}\}} z_{itk} \leq r_{it}, \ \forall \ (i, t) \in \mathcal{I} \times \mathcal{T}. \tag{49}$$

Such inequalities are also unnecessary when $\bar{K}_{it} < \bar{G}_{it}$.

3. From (45) we notice that

$$\bar{G}_{it} \leq \underline{K}_{it} + \bar{R}_{it}, \quad (i,t) \in \mathcal{I} \times \mathcal{T}. \tag{50}$$

Then, applying (50) on the left hand side of (6) gives

$$r_{it} + \sum_{k \in \mathcal{K}(i,t)} k z_{itk} \leq \underline{K}_{it} + \bar{R}_{it}, \ \forall \ (i,t) \in \mathcal{I} \times \mathcal{T},$$

which by (7) and (44) leads to

$$r_{it} + \sum_{k \in \mathcal{K}(i,t) \setminus \{\underline{K}_{it}\}} (k - \underline{K}_{it}) z_{itk} \leq \bar{R}_{it}, \ \forall \ (i,t) \in \mathcal{I}, \times \mathcal{T}. \tag{51}$$

A.4 Proof of Proposition 2

Proof. Recall that we assume $B_t^- \geq B_t^+ \geq 0 \ \ \forall t \in \mathcal{T}$. By (16),

$$\begin{aligned} \sum_{\substack{i \in \mathcal{I}, \\ k \in \mathcal{K}(i,t)}} z_{itk} \bar{P}_{ik} &\geq \sum_{\substack{i \in \mathcal{I}, \\ k \in \mathcal{K}(i,t)}} p_{itk\omega}, \forall (t,\omega) \in \mathcal{T} \times \Omega \\ &= \sum_{i \in \mathcal{I}} p_{it\omega}, \forall (t,\omega) \in \mathcal{T} \times \Omega \qquad \text{[by (17)]}. \end{aligned} \tag{52}$$

Then from (52),

$$\begin{aligned} \sum_{t \in \mathcal{T}} B_t^+ \Big(\sum_{\substack{i \in \mathcal{I} \\ k \in \mathcal{K}(i,t)}} z_{itk} \bar{P}_{ik} - A_t \Big) &\geq \sum_{t \in \mathcal{T}} B_t^+ \Big(\sum_{i \in I} p_{it\omega} - A_t \Big), \ \forall \omega \in \Omega \quad [B_t^+ \geq 0] \\ &= \sum_{t \in \mathcal{T}} B_t^+ \Big(q_{t\omega}^+ - q_{t\omega}^- \Big), \forall \omega \in \Omega \quad \text{[by (18)]}, \\ &\geq \sum_{t \in \mathcal{T}} (B_t^+ q_{t\omega}^+ - B_t^- q_{t\omega}^-), \forall \omega \in \Omega \quad [B_t^- \geq B_t^+] \end{aligned} \tag{53}$$

Multiplying each side of (53) by the scenario probability φ_ω, and summing over the elements of Ω yields,

$$\sum_{t \in \mathcal{T}} B_t^+ \Big(\sum_{k \in \mathcal{K}(i,t)} z_{itk} \bar{P}_{ik} - A_t \Big) \geq \sum_{\substack{t \in \mathcal{T} \\ \omega \in \Omega}} \varphi_\omega (B_t^+ q_{t\omega}^+ - B_t^- q_{t\omega}^-), \tag{54}$$

where the left-hand side of (54) is simplified due to $\sum_{\omega \in \Omega} \varphi_\omega = 1$. Since the right-hand side of (54) is the expected value of (21) over the set of scenarios Ω, the left-hand side of (54) is a valid upper bound on the expected optimal value of the subproblem. $\square$

Appendix B: Selecting multiple-phase relaxation sequence and valid inequalities

B.1 Valid Inequalities

On the set of 24 instances, we ran a factorial experiment with the $2^3 = 8$ combinations of the three families of valid inequalities of Section 5.1.5. To select the best combination of these inequalities, we sequentially applied analysis of variance (ANOVA) with normalized computational time as the response variable. From the results of the first ANOVA, with each family of valid inequalities defined as a categorical factor (Table 8), we dropped the family 1 of valid inequalities (factor VI1) for increasing the computational times ($\beta = 0.188$) at a significance level of 0.1 (p-value = 0.055). With the same experimental data, an ANOVA

Table 8: Summary of ANOVA with valid inequalities 1, 2 and 3 as main factors, and normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.078	0.427
VI1	0.188	0.055
VI2	−0.095	0.333
VI3	−0.249	0.011

with the factors VI2 and VI3 and the interaction term VI2·VI3 (see Table 9) shows that the combination of the valid inequalities 2 and 3 (i.e., the interaction term VI2·V3) has the lowest average computational time ($\beta = -0.363$), at a significance level of 0.1 (p-value = 0.064).

B.2 Multiple-phase relaxation

We defined the relaxation sequences of Table 2 as treatments. In these sequences, each phase is completed when either a specified maximum number of cuts or a maximum optimality gap is reached (Table 10). According to the results, the sequence without relaxation (i.e., relaxation sequence 0), exhibited

Table 9: Summary of ANOVA with valid inequalities 1 and 2 and interaction term, and normalized computational time as response variable.

	β estimate	p-value
(Intercept)	0.081	0.408
VI2	0.087	0.531
VI3	−0.067	0.627
VI2·VI3	−0.363	0.064

Table 10: Parameters of stages in multi-phase relaxation.

Relax. level	Binary var.	Max. cuts	Max. gap
0	y, z	1000	1.0e-5
1	y	4	0.005
2	z	4	0.005
3	-	20	0.010

the largest variability and the highest computational time (Fig. 10). An analysis of variance on the 24 instances indicated that the multi-phase relaxation had a significant effect on the computational times (p-value = 0.00924). Although the computational times of the relaxation sequences 3, 4 and 5 were similar, the relaxation sequence 4 showed the most significant effect (p-value = 0.007) in a one-tailed t-test against the method without relaxation (see Table 11). Therefore, the best configuration applies the relaxation sequence $(y, z) \rightarrow (z) \rightarrow (y)$, before solving the master problem without relaxation.

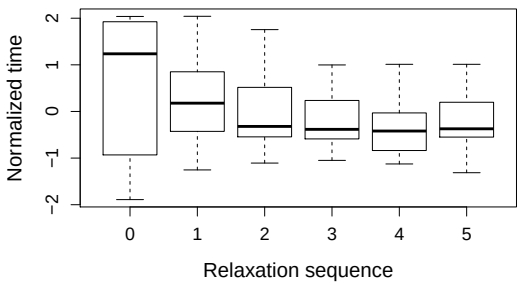


Figure 10: Boxplot of the computational times of the multi-phase relaxation sequences on 24 instances.

Table 11: Summary statistics of normalized computational times of multi-phase relaxations. The column *Diff.* shows the difference between the mean time of each sequence and the mean of sequence 0.

Relax. Seq.	Mean	St. Dev.	Diff.	p-value
0	0.54	1.51	0.00	-
1	0.21	0.91	−0.33	0.181
2	−0.04	0.71	−0.58	0.048
3	−0.19	0.57	−0.73	0.017
4	−0.34	0.63	−0.88	0.007
5	−0.18	0.60	−0.72	0.019

Appendix C: Nomenclature

Primary sets	
$\mathcal{I}$	Powerhouses
$\mathcal{M}$	Maintenance tasks
$\mathcal{T}$	Planning time periods, $t \in \mathcal{T} = \{1 \dots T\}$
Ω	Scenarios
Parameters	
$\xi_{it\omega}$	Lateral inflows to powerhouse i in period t and scenario ω, [m^3/s].
A_t	Electricity load at time period t.
B_t^+	Electricity sale price in time period t, [$/MWh].
B_t^-	Electricity purchase price in time period t, [$/MWh].
C_{mt}	Total cost of maintenance task m started at time period t, [$].
D_m	Duration of maintenance task m [day].
E_m	Earliest start time period of maintenance task m.
F	Factor for conversion from flow per second in m^3 to flow per day in hm^3 [0.0864·s·hm^3 ·/(day'·m^3)].
$\bar{G}_{it}$	Maximum number of available turbines in powerhouse i at time period t, [turbines].
$\underline{G}_i$	Minimum number of available turbines in powerhouse i [turbines].
L_m	Latest start time period of maintenance task m.
O_{it}	Maximum number of turbine outages in powerhouse i at time period t, [turbines].
$\bar{P}_i$	Generation capacity in powerhouse i, [MWh/day].
$\bar{P}_{ik}$	Generation capacity in powerhouse i when k turbines are active, [MWh/day].
$Q(\bar{y})$	Expected operating cost of solution $\bar{y}$ [$].
$Q_\omega(\bar{y})$	Expected operating cost of solution $\bar{y}$ in scenario ω [$].
$\bar{R}_{it}$	Number of maintenance activities that *can* be in execution at powerhouse i in time period t.
$\underline{R}_{it}$	Number of maintenance activities that *must* be in execution at powerhouse i in time period t.
S_{0i}	Initial volume in reservoir of powerhouse i, [hm^3].
$\underline{S}_i, \bar{S}_i$	Limits on stored water in reservoir of powerhouse i at period t [hm^3].
$\bar{U}_{it}$	Maximum discharge rate in powerhouse i, [m^3/s].
$\bar{V}_{it}$	Maximum water spill in powerhouse i, [m^3/s].
$\bar{W}_t^+$	Maximum electricity sale at time t [MWh/day].
$\bar{W}_t^-$	Maximum electricity purchase at time t [MWh/day].
Derived sets	
$\mathcal{T}(m)$	Time periods when maintenance task m can be initiated in order to be completed within $\mathcal{T}$.
$\mathcal{M}(i)$	Maintenance tasks m that should be executed in powerhouse i.
$\mathcal{M}(i,t)$	Maintenance tasks m that can be in execution in powerhouse i at time period t.
$\mathcal{U}(i)$	Powerhouses upstream of powerhouse i ($\mathcal{U}(i) \subset \mathcal{I}$).
$\mathcal{K}(i,t)$	Numbers of generators that can be active at time period t and powerhouse i.
$\mathcal{H}(i,k)$	Hyperplanes for approximating the maximum power of powerhouse i when k turbines are active.
$\mathcal{A}$	set of indices (m,t) of variables y_{mt} with value 1 in solution $\bar{y}$, i.e, $\mathcal{A} = \{(m,t) \in \mathcal{M} \times \mathcal{T} \mid \bar{y}_{mt} = 1\}$.

Parameters with indexes in derived sets	
β_h^u	Coefficient of u_{it} in hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh· s/(m^3·day)].
β_h^s	Coefficient of s_{it} in hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh/(hm^3·day)].
β_h^0	Independent term of hyperplane $h \in \mathcal{H}(i,k)$ for bounding the power output of powerhouse i when k generators are active [MWh/day].
Decision variables	
$p_{it\omega}$	Generation of powerhouse i during time period t in scenario ω [MWh/day].
$p_{itk\omega}$	Generation of powerhouse i during time period t in scenario ω when k generators are active [MWh/day].
$q_{t\omega}^+$	Sale of electricity at period t in scenario ω [MWh].
$q_{t\omega}^-$	Purchase of electricity at period t in scenario ω [MWh].
r_{it}	Number of maintenance activities in execution at powerhouse i and time period t.
$s_{it\omega}$	Content of reservoir in powerhouse i at the end of period t in scenario ω [hm^3].
$u_{it\omega}$	Water discharge of turbines in powerhouse i at time period t in scenario ω [m^3/s].
$v_{it\omega}$	Water spill of reservoir in powerhouse i at time period t in scenario ω [m^3/s].
y_{mt}	Binary variable with value 1 if maintenance task m initiates at time period t, 0 otherwise.
z_{itk}	Binary variable with value 1 if k hydro-turbines are active in powerhouse i at time t, 0 otherwise.
z^{SP}	Approximated expected profit of the hydroelectric production [$].
z_ω^{SP}	Profit of the hydroelectric production in scenario ω [$].
Dual variables	
$\pi_{it\omega}^p$	of mass balance constraint (14) in solution p.
$\gamma_{itkh\omega}^p$	of power function (15) in solution p.
$\lambda_{itk\omega}^p$	of power bound constraint (16) in solution p.
$\psi_{t\omega}^p$	of power balance constraint (18) in solution p.
$\theta_{it\omega}^p$	of sum of power constraint (17) in solution p.
Reduced costs	
$\alpha_{it\omega}^{pu}$	of water discharge variable $u_{it\omega}$ in solution p
$\alpha_{it\omega}^{ps}$	of water level variable $s_{it\omega}$ in solution p
$\alpha_{t\omega}^{p+}$	of electricity sale variable $q_{t\omega}^+$ in solution p
$\alpha_{t\omega}^{p-}$	of electricity purchase variable $q_{t\omega}^-$ in solution p